



1  
2  
3  
4

**Document Number: DSP0804**

**Date: 2009-06-04**

**Version: 1.0.0**

5 **Modular Systems Profile SM CLP Command**  
6 **Mapping Specification**

7 **Document Type: Specification**  
8 **Document Status: DMTF Standard**  
9 **Document Language: E**

10

11 Copyright notice

12 Copyright © 2006, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

13 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
14 management and interoperability. Members and non-members may reproduce DMTF specifications and  
15 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to  
16 time, the particular version and release date should always be noted.

17 Implementation of certain elements of this standard or proposed standard may be subject to third party  
18 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations  
19 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,  
20 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or  
21 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to  
22 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,  
23 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or  
24 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any  
25 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent  
26 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is  
27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  
28 implementing the standard from any and all claims of infringement by a patent owner for such  
29 implementations.

30 For information about patents held by third-parties which have notified the DMTF that, in their opinion,  
31 such patent may relate to or impact implementations of DMTF standards, visit  
32 <http://www.dmtf.org/about/policies/disclosures.php>.

33

## CONTENTS

34	Foreword .....	5
35	Introduction .....	6
36	1 Scope .....	7
37	2 Normative References.....	7
38	2.1 Approved References .....	7
39	2.2 Other References.....	7
40	3 Terms and Definitions.....	7
41	4 Symbols and Abbreviated Terms.....	8
42	5 Recipes.....	9
43	6 Mappings.....	9
44	6.1 CIM_AdminDomain.....	9
45	6.2 CIM_ComputerSystem.....	11
46	6.3 CIM_ConcreteDependency .....	13
47	6.4 CIM_SystemComponent.....	15
48	ANNEX A (informative) Change Log .....	20
49		

## 50 Tables

51	Table 1 – Command Verb Requirements for CIM_AdminDomain.....	9
52	Table 2 – Command Verb Requirements for CIM_ComputerSystem .....	11
53	Table 3 – Command Verb Requirements for CIM_ConcreteDependency .....	13
54	Table 4 – Command Verb Requirements for CIM_SystemComponent.....	16
55		



57

## Foreword

58 The *Modular Systems Profile SM CLP Command Mapping Specification* (DSP0804) was prepared by the  
59 Server Management Working Group.

### 60 **Conventions**

61 The pseudo-code conventions utilized in this document are the Recipe Conventions as defined in SNIA  
62 [SMI-S 1.1.0](#), section 7.6.

### 63 **Acknowledgements**

64 The authors wish to acknowledge the following participants from the DTMF Server Management Working  
65 Group:

- 66 • Aaron Merkin – IBM
- 67 • Jon Hass – Dell
- 68 • Khachatur Papanyan – Dell
- 69 • Jeff Hilland – HP
- 70 • Christina Shaw – HP
- 71 • Perry Vincent – Intel
- 72 • John Leung – Intel

73

74

## Introduction

75 This document defines the SM CLP mapping for CIM elements described in the [Modular Systems Profile](#).  
76 The information in this specification, combined with the [SM CLP-to-CIM Common Mapping Specification](#)  
77 [1.0](#), is intended to be sufficient to implement SM CLP commands relevant to the classes, properties, and  
78 methods described in the [Modular Systems Profile](#) using CIM operations.

79 The target audience for this specification is implementers of the SM CLP support for [Modular Systems](#)  
80 [Profile](#).



- 112 **3.4**  
113 **mandatory**  
114 indicates requirements to be followed strictly in order to conform to the document and from which no  
115 deviation is permitted
- 116 **3.5**  
117 **may**  
118 indicates a course of action permissible within the limits of the document
- 119 **3.6**  
120 **need not**  
121 indicates a course of action permissible within the limits of the document
- 122 **3.7**  
123 **optional**  
124 indicates a course of action permissible within the limits of the document
- 125 **3.8**  
126 **shall**  
127 indicates requirements to be followed strictly in order to conform to the document and from which no  
128 deviation is permitted
- 129 **3.9**  
130 **shall not**  
131 indicates requirements to be followed strictly in order to conform to the document and from which no  
132 deviation is permitted
- 133 **3.10**  
134 **should**  
135 indicates that among several possibilities, one is recommended as particularly suitable, without  
136 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 137 **3.11**  
138 **should not**  
139 indicates that a certain possibility or course of action is deprecated but not prohibited

## 140 **4 Symbols and Abbreviated Terms**

141 The following symbols and abbreviations are used in this document.

- 142 **4.1**  
143 **CIM**  
144 Common Information Model
- 145 **4.2**  
146 **CLP**  
147 Command Line Protocol
- 148 **4.3**  
149 **DMTF**  
150 Distributed Management Task Force



151 **4.4**  
 152 **IETF**  
 153 Internet Engineering Task Force

154 **4.5**  
 155 **SM**  
 156 Server Management

157 **4.6**  
 158 **SMI-S**  
 159 Storage Management Initiative Specification

160 **4.7**  
 161 **SNIA**  
 162 Storage Networking Industry Association

## 163 **5 Recipes**

164 The following is a list of the common recipes used by the mappings in this specification. For a definition of  
 165 each recipe, see the *SM CLP-to-CIM Common Mapping Specification 1.0* ([DSP0216](#)).

- 166 • smShowInstance
- 167 • smShowInstances
- 168 • smShowAssociationInstance
- 169 • smShowAssociationInstances

## 170 **6 Mappings**

171 The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in  
 172 the [Modular Systems Profile](#). Requirements specified here related to the support for a CLP verb for a  
 173 particular class are solely within the context of this profile.

### 174 **6.1 CIM\_AdminDomain**

175 The `cd` and `help` verbs shall be supported as described in [DSP0216](#).

176 Table 1 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of  
 177 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the  
 178 verb and target. Table 1 is for informational purposes only; in case of a conflict between Table 1 and  
 179 requirements detailed in the following sections, the text detailed in the following sections supersedes the  
 180 information in Table 1.

181 **Table 1 – Command Verb Requirements for CIM\_AdminDomain**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	

Command Verb	Requirement	Comments
set	Not supported	
show	Shall	See 6.1.2.
start	Not supported	
stop	Not supported	

182 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,  
183 reset, start, and stop.

### 184 6.1.1 Ordering of Results

185 When results are returned for multiple instances of CIM\_AdminDomain, implementations shall utilize the  
186 following algorithm to produce the natural (that is, default) ordering:

- 187 • Results for CIM\_AdminDomain are unordered; therefore, no algorithm is defined.

### 188 6.1.2 Show

189 This section describes how to implement the `show` verb when applied to an instance of  
190 CIM\_AdminDomain. Implementations shall support the use of the `show` verb with CIM\_AdminDomain.

191 The `show` verb is used to display information about the management domain.

#### 192 6.1.2.1 Show a Single Instance

193 This command form is for the `show` verb applied to a single instance of CIM\_AdminDomain.

##### 194 6.1.2.1.1 Command Form

```
195 show <CIM_AdminDomain single instance>
```

##### 196 6.1.2.1.2 CIM Requirements

197 See CIM\_AdminDomain in the "CIM\_AdminDomain – Power Domain" and "CIM\_AdminDomain – Cooling  
198 Domain" sections of the [Modular Systems Profile](#) for the list of mandatory properties.

##### 199 6.1.2.1.3 Behavior Requirements

###### 200 6.1.2.1.3.1 Preconditions

201 In this section, `$instance` represents the targeted instance of CIM\_AdminDomain.

```
202 $instance=<CIM_AdminDomain single instance>;
```

203 `#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

###### 204 6.1.2.1.3.2 Pseudo Code

```
205 #propertylist[] = NULL;
206 if ( false == #all)
207 {
208     #propertylist[] = <array of mandatory non-key property names (see CIM
209     Requirements)>;
210 }
211 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
212 &smEnd;
```

213 **6.1.2.2 Show Multiple Instances**

214 This command form is for the `show` verb applied to multiple instances of `CIM_AdminDomain`. Each  
 215 instance of `CIM_AdminDomain` is scoped to a `CIM_ComputerSystem` instance through the  
 216 `CIM_SystemComponent` association.

217 **6.1.2.2.1 Command Form**

```
218 show <CIM_AdminDomain multiple instances>
```

219 **6.1.2.2.2 CIM Requirements**

220 See `CIM_AdminDomain` in the “`CIM_AdminDomain – Power Domain`” and “`CIM_AdminDomain – Cooling`  
 221 `Domain`” sections of the [Modular Systems Profile](#) for the list of mandatory properties.

222 **6.1.2.2.3 Behavior Requirements**

223 **6.1.2.2.3.1 Preconditions**

224 In this section, `$containerInstance` represents the instance of `CIM_ComputerSystem` which  
 225 represents the container system and is associated to the targeted instances of `CIM_AdminDomain`  
 226 through the `CIM_SystemComponent` association.

227 `#all` is true if the “all” option was specified with the command; otherwise, `#all` is false.

228 **6.1.2.2.3.2 Pseudo Code**

```
229 #propertylist[] = NULL;  

    230 if ( false == #all)  

    231 {  

    232     #propertylist[] = <array of mandatory non-key property names (see CIM  

    233         Requirements)>;  

    234 }  

    235 &smShowInstances ( "CIM_AdminDomain", "CIM_SystemComponent",  

    236     $containerInstance.getObjectPath(), "GroupComponent", "PartComponent",  

    237     #propertylist[] );  

    238 &smEnd;
```

239 **6.2 CIM\_ComputerSystem**

240 The `cd`, `exit`, `help`, and `version` verbs shall be supported as described in [DSP0216](#).

241 Table 2 lists each SM CLP verb, the required level of support for the verb in conjunction with the target  
 242 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and  
 243 target. Table 2 is for informational purposes only; in case of a conflict between Table 2 and requirements  
 244 detailed in the following sections, the text detailed in the following sections supersedes the information in  
 245 Table 2.

246 **Table 2 – Command Verb Requirements for `CIM_ComputerSystem`**

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	

Command Verb	Requirement	Comments
Set	Not supported	
Show	Shall	See 6.2.2.
Start	Not supported	
Stop	Not supported	

247 No mapping is defined for the following verbs for the specified target: create, delete, dump, and load.

## 248 6.2.1 Ordering of Results

249 When results are returned for multiple instances of CIM\_ComputerSystem, implementations shall utilize  
250 the following algorithm to produce the natural (that is, default) ordering:

- 251 • Results for CIM\_ComputerSystem are unordered; therefore, no algorithm is defined.

## 252 6.2.2 Show

253 This section describes how to implement the `show` verb when applied to an instance of  
254 CIM\_ComputerSystem. Implementations shall support the use of the `show` verb with  
255 CIM\_ComputerSystem.

### 256 6.2.2.1 Show Command Form for Multiple Instances Target

257 This command form is used to show many instances of CIM\_ComputerSystem.

#### 258 6.2.2.1.1 Command Form

```
259 show <CIM_ComputerSystem multiple instances>
```

#### 260 6.2.2.1.2 CIM Requirements

261 See CIM\_ComputerSystem in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
262 mandatory properties.

#### 263 6.2.2.1.3 Behavior Requirements

##### 264 6.2.2.1.3.1 Preconditions

265 In this section, `$containerInstance` represents the instance of a subclass of CIM\_System which  
266 represents the container system and is associated to the targeted instances of CIM\_ComputerSystem  
267 through the CIM\_SystemComponent association.

268 `#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

##### 269 6.2.2.1.3.2 Pseudo Code

```
270 #propertylist[] = NULL;
271 if ( false == #all)
272 {
273     #propertylist[] = <array of mandatory non-key property names (see CIM
274         Requirements)>;
275 }
276 &smShowInstances ( "CIM_ComputerSystem", "CIM_SystemComponent",
277     $containerInstance.getObjectPath(), #propertylist[] );
278 &smEnd;
```

279 **6.2.2.2 Show Command Form for a Single Instance Target**

280 This command form is used to show a single instance of CIM\_ComputerSystem.

281 **6.2.2.2.1 Command Form**

282 `show <CIM_ComputerSystem single instance>`

283 **6.2.2.2.1.1 CIM Requirements**

284 See CIM\_ComputerSystem in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
285 mandatory properties.

286 **6.2.2.2.2 Behavior Requirements**

287 **6.2.2.2.2.1 Preconditions**

288 In this section, \$instance represents the targeted instance of CIM\_ComputerSystem.

289 `$instance=<CIM_ComputerSystem single instance>;`

290 #all is true if the “-all” option was specified with the command; otherwise, #all is false.

291 **6.2.2.2.2.2 Pseudo Code**

```
292 #propertylist[] = NULL;
293 if ( false == #all)
294 {
295     #propertylist[] = <array of mandatory non-key property names (see CIM
296     Requirements)>;
297 }
298 &smShowInstance ( $instance.getObjectPath(), #propertylist[] );
299 &smEnd;
```

300 **6.3 CIM\_ConcreteDependency**

301 The cd and help verbs shall be supported as described in [DSP0216](#).

302 Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of  
303 the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the  
304 verb and target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and  
305 requirements detailed in the following sections, the text detailed in the following sections supersedes the  
306 information in Table 3.

307 **Table 3 – Command Verb Requirements for CIM\_ConcreteDependency**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.3.2.

Command Verb	Requirement	Comments
start	Not supported	
stop	Not supported	

308 No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`,  
 309 `reset`, `set`, `start`, and `stop`.

### 310 6.3.1 Ordering of Results

311 When results are returned for multiple instances of `CIM_ConcreteDependency`, implementations shall  
 312 utilize the following algorithm to produce the natural (that is, default) ordering:

- 313 • Results for `CIM_ConcreteDependency` are unordered; therefore, no algorithm is defined.

### 314 6.3.2 Show

315 This section describes how to implement the `show` verb when applied to an instance of  
 316 `CIM_ConcreteDependency`. Implementations shall support the use of the `show` verb with  
 317 `CIM_ConcreteDependency`.

318 The `show` command is used to display information about the `CIM_ConcreteDependency` instance or  
 319 instances.

#### 320 6.3.2.1 Show Multiple Instances – `CIM_PhysicalPackage` Reference

321 This command form is for the `show` verb applied to multiple instances. This command form corresponds  
 322 to a `show` command issued against `CIM_ConcreteDependency` where only one reference is specified and  
 323 the reference is to an instance of `CIM_PhysicalPackage` representing a blade package.

##### 324 6.3.2.1.1 Command Form

```
325 show <CIM_ConcreteDependency multiple instances>
```

##### 326 6.3.2.1.2 CIM Requirements

327 See `CIM_ConcreteDependency` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list  
 328 of mandatory properties.

##### 329 6.3.2.1.3 Behavior Requirements

###### 330 6.3.2.1.3.1 Preconditions

331 `$instance` contains the instance of `CIM_PhysicalPackage` which is referenced by  
 332 `CIM_ConcreteDependency`.

###### 333 6.3.2.1.3.2 Pseudo Code

```
334 &smShowAssociationInstances ( "CIM_ConcreteDependency", $instance.getObjectPath() );  
335 &smEnd;
```

#### 336 6.3.2.2 Show a Single Instance – `CIM_PhysicalPackage` Reference

337 This command form is for the `show` verb applied to a single instance. This command form corresponds to  
 338 a `show` command issued against `CIM_ConcreteDependency` where the reference specified is to an  
 339 instance of `CIM_PhysicalPackage` representing a blade expansion. An instance of `CIM_PhysicalPackage`

340 representing a blade expansion is referenced by exactly one instance of CIM\_ConcreteDependency.  
 341 Therefore, a single instance will be returned.

### 342 6.3.2.2.1 Command Form

```
343 show <CIM_ConcreteDependency single instance>
```

### 344 6.3.2.2.2 CIM Requirements

345 See CIM\_ConcreteDependency in the “CIM Elements” section of the [Modular Systems Profile](#) for the list  
 346 of mandatory properties.

### 347 6.3.2.2.3 Behavior Requirements

#### 348 6.3.2.2.3.1 Preconditions

349 \$instance contains the instance of CIM\_PhysicalPackage representing a blade expansion that is  
 350 referenced by CIM\_ConcreteDependency.

#### 351 6.3.2.2.3.2 Pseudo Code

```
352 &smShowAssociationInstances ( "CIM_ConcreteDependency", $instance.getObjectPath() );  
353 &smEnd;
```

### 354 6.3.2.3 Show a Single Instance – Both References

355 This command form is for the show verb applied to a single instance. This command form corresponds to  
 356 a show command issued against CIM\_ConcreteDependency where both references are specified and  
 357 therefore the desired instance is unambiguously identified.

#### 358 6.3.2.3.1 Command Form

```
359 show <CIM_ConcreteDependency single instance>
```

#### 360 6.3.2.3.2 CIM Requirements

361 See CIM\_ConcreteDependency in the “CIM Elements” section of the [Modular Systems Profile](#) for the list  
 362 of mandatory properties.

#### 363 6.3.2.3.3 Behavior Requirements

##### 364 6.3.2.3.3.1 Preconditions

365 \$instanceA contains the instance of CIM\_PhysicalPackage representing a blade that is referenced by  
 366 CIM\_ConcreteDependency.

367 \$instanceB contains the instance of CIM\_PhysicalPackage representing a blade expansion that is  
 368 referenced by CIM\_ConcreteDependency.

##### 369 6.3.2.3.3.2 Pseudo Code

```
370 &smShowAssociationInstance ( "CIM_ConcreteDependency", $instanceA.getObjectPath(),  
371     $instanceB.getObjectPath() );  
372 &smEnd;
```

## 373 6.4 CIM\_SystemComponent

374 The cd, exit, help, and version verbs shall be supported as described in [DSP0216](#).

375 Table 4 lists each SM CLP verb, the required level of support for the verb in conjunction with the target  
 376 class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and  
 377 target. Table 4 is for informational purposes only; in case of a conflict between Table 4 and requirements  
 378 detailed in the following sections, the text detailed in the following sections supersedes the information in  
 379 Table 4.

380 **Table 4 – Command Verb Requirements for CIM\_SystemComponent**

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.4.2.
Start	Not supported	
Stop	Not supported	

381 No mapping is defined for the following verbs for the specified target: create, delete, dump, load,  
 382 reset, set, start, and stop.

### 383 6.4.1 Ordering of Results

384 When results are returned for multiple instances of CIM\_SystemComponent, implementations shall utilize  
 385 the following algorithm to produce the natural (that is, default) ordering:

- 386 • Results for CIM\_SystemComponent are unordered; therefore, no algorithm is defined.

### 387 6.4.2 Show

388 This section describes how to implement the `show` verb when applied to an instance of  
 389 CIM\_SystemComponent. Implementations shall support the use of the `show` verb with  
 390 CIM\_SystemComponent.

#### 391 6.4.2.1 Show Command Form for Multiple Instances – Modular System Reference

392 This command form is used to show many instances of CIM\_SystemComponent. This command form  
 393 corresponds to a `show` command issued against the instance of CIM\_SystemComponent where only one  
 394 reference is specified and the reference is to an instance of CIM\_ComputerSystem that represents a  
 395 modular system.

##### 396 6.4.2.1.1 Command Form

397 `show <CIM_SystemComponent multiple instances>`

##### 398 6.4.2.1.2 CIM Requirements

399 See CIM\_SystemComponent in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
 400 mandatory properties.



### 401 6.4.2.1.3 Behavior Requirements

#### 402 6.4.2.1.3.1 Preconditions

403 In this section, `$instance` represents the instance of a `CIM_ComputerSystem` which is referenced by  
404 `CIM_SystemComponent`.

#### 405 6.4.2.1.3.2 Pseudo Code

```
406 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.GetObjectPath() );  
407 &smEnd;
```

### 408 6.4.2.2 Show Command Form for Multiple Instances – Chassis Manager Reference

409 This command form is used to show a single instance of `CIM_SystemComponent`. This command form  
410 corresponds to a `show` command issued against a single instance of `CIM_SystemComponent`, where  
411 only one reference is specified and the reference is to the instance of `CIM_ComputerSystem` that  
412 represents a Chassis Manager. `CIM_SystemComponent` will associate the `CIM_ComputerSystem`  
413 instance for the chassis manager to the modular system and could associate it to one or more  
414 `CIM_AdminDomain` instances.

#### 415 6.4.2.2.1 Command Form

```
416 show <CIM_SystemComponent multiple instances>
```

#### 417 6.4.2.2.2 CIM Requirements

418 See `CIM_SystemComponent` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
419 mandatory properties.

### 420 6.4.2.2.3 Behavior Requirements

#### 421 6.4.2.2.3.1 Preconditions

422 In this section `$instance` represents the instance of `CIM_ComputerSystem` which is referenced by  
423 `CIM_SystemComponent`.

#### 424 6.4.2.2.3.2 Pseudo Code

```
425 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.GetObjectPath() );  
426 &smEnd;
```

### 427 6.4.2.3 Show Command Form for Multiple Instances– Processor Blade Reference

428 This command form is used to show a single instance of `CIM_SystemComponent`. This command form  
429 corresponds to a `show` command issued against a single instance of `CIM_SystemComponent`, where  
430 only one reference is specified and the reference is to the instance of `CIM_ComputerSystem` that  
431 represents a Processor Blade. `CIM_SystemComponent` will associate the `CIM_ComputerSystem`  
432 instance for the processor blade to the modular system and could associate it to one or more  
433 `CIM_AdminDomain` instances.

#### 434 6.4.2.3.1 Command Form

```
435 show <CIM_SystemComponent multiple instances>
```

#### 436 6.4.2.3.2 CIM Requirements

437 See `CIM_SystemComponent` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
438 mandatory properties.

### 439 6.4.2.3.3 Behavior Requirements

#### 440 6.4.2.3.3.1 Preconditions

441 In this section `$instance` represents the instance of `CIM_ComputerSystem` which is referenced by  
442 `CIM_SystemComponent`.

#### 443 6.4.2.3.3.2 Pseudo Code

```
444 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.GetObjectPath() );  
445 &smEnd;
```

### 446 6.4.2.4 Show Command Form for Multiple Instances – Cooling Domain Reference

447 This command form is used to show a single instance of `CIM_SystemComponent`. This command form  
448 corresponds to a `show` command issued against a single instance of `CIM_SystemComponent`, where  
449 only one reference is specified and the reference is to the instance of `CIM_AdminDomain` that represents  
450 a cooling domain. `CIM_SystemComponent` will associate the `CIM_AdminDomain` instance for the cooling  
451 domain to the modular system and could associate it to one or more contained  
452 `CIM_ManagedSystemElement` instances.

#### 453 6.4.2.4.1 Command Form

```
454 show <CIM_SystemComponent single instance>
```

#### 455 6.4.2.4.2 CIM Requirements

456 See `CIM_SystemComponent` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
457 mandatory properties.

### 458 6.4.2.4.3 Behavior Requirements

#### 459 6.4.2.4.3.1 Preconditions

460 In this section, `$instance` represents the instance of `CIM_AdminDomain` which is referenced by  
461 `CIM_SystemComponent`.

#### 462 6.4.2.4.3.2 Pseudo Code

```
463 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.GetObjectPath() );  
464 &smEnd;
```

### 465 6.4.2.5 Show Command Form for Multiple Instances – Power Domain Reference

466 This command form is used to show a single instance of `CIM_SystemComponent`. This command form  
467 corresponds to a `show` command issued against a single instance of `CIM_SystemComponent`, where  
468 only one reference is specified and the reference is to the instance of `CIM_AdminDomain` that represents  
469 a power domain. `CIM_SystemComponent` will associate the `CIM_AdminDomain` instance for the power  
470 domain to the modular system and could associate it to one or more contained  
471 `CIM_ManagedSystemElement` instances.

#### 472 6.4.2.5.1 Command Form

```
473 show <CIM_SystemComponent single instance>
```

#### 474 6.4.2.5.2 CIM Requirements

475 See `CIM_SystemComponent` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
476 mandatory properties.

### 477 6.4.2.5.3 Behavior Requirements

#### 478 6.4.2.5.3.1 Preconditions

479 In this section, `$instance` represents the instance of `CIM_AdminDomain` which is referenced by  
480 `CIM_SystemComponent`.

#### 481 6.4.2.5.3.2 Pseudo Code

```
482 &smShowAssociationInstances ( "CIM_SystemComponent", $instance.GetObjectPath() );  
483 &smEnd;
```

### 484 6.4.2.6 Show Command Form for a Single Instance Target – Both References

485 This command form is for the `show` verb applied to a single instance. This command form corresponds to  
486 a `show` command issued against `CIM_SystemComponent` where both references are specified and  
487 therefore the desired instance is unambiguously identified.

#### 488 6.4.2.6.1 Command Form

```
489 show <CIM_SystemComponent single instance>
```

#### 490 6.4.2.6.2 CIM Requirements

491 See `CIM_SystemComponent` in the “CIM Elements” section of the [Modular Systems Profile](#) for the list of  
492 mandatory properties.

### 493 6.4.2.6.3 Behavior Requirements

#### 494 6.4.2.6.3.1 Preconditions

495 In this section, `$instanceA` represents the instance of `CIM_ManagedSystemElement` referenced by the  
496 `CIM_SystemComponent` association. `$instanceB` represents the instance of a subclass of `CIM_System`  
497 that is referenced by `CIM_SystemComponent`.

#### 498 6.4.2.6.3.2 Pseudo Code

```
499 &smShowAssociationInstance ( "CIM_SystemComponent", $instanceA.GetObjectPath(),  
500     $instanceB.GetObjectPath() );  
501 &smEnd;
```

502

503  
504  
505  
506  
507

**ANNEX A**  
(informative)

**Change Log**

Version	Date	Author	Description
1.0.0	2009-06-04		DMTF Standard Release

508